
From μ -Calculus to Alternating Tree Automata using Parity Games

M. Fareed Arif

February 3, 2016

Abstract μ -Calculus and automata on infinite trees are complementary ways of describing infinite tree languages. The correspondence between μ -Calculus and alternating tree automaton is used to solve the satisfiability and model checking problems by compiling the modal μ -Calculus formula into an alternating tree automata. Thus advocating an automaton model specially tailored for working with modal μ -Calculus. The advantage of the automaton model is its ability to deal with arbitrary branching in a much simpler way as compare to the one proposed by Janin and Walukiewicz. Both problems (i.e., model checking and satisfiability) are solved by reduction to the corresponding problems of alternating tree automata, namely to the acceptance and the non-emptiness problems, respectively. These problems, in turn, are solved using parity games where semantics of alternating tree automata is translated to a winning strategy in an appropriate parity game.

Keywords Game Playing · μ -Calculus · Alternating Tree Automaton · Parity Games

1 Introduction

There is a growing recognition to apply formal mathematical methods for specifying and reasoning about the correctness of computing systems namely in model checking. Model checking presents an efficient and more expressive method to automatically verify a transition system whether it meets a correctness specification formulated in μ -Calculus. Modal μ -Calculus is a fundamental logic for specifying properties of transition systems. It is a quite expressive language that subsumes most of the common logical formalisms used in verification including LTL, CTL, CTL* and PDL [4]. The μ -Calculus over binary trees coincide in expressive power with alternating tree automata and it is as expressive as Monadic Second-order logic (MSOL) on trees [6].

F. Author
CASL, University College Dublin, Ireland,
Tel.: +353 (1) 7162641
Fax: +353 (1) 7165396
E-mail: 12250832@ucdconnect.ie

In this paper, we explain a method to translate a given μ -Calculus formula φ into an appropriate finite automaton \mathcal{A} such that $L(\varphi) = L(\mathcal{A})$ [7]. Such a translation reduces the model checking and the satisfiability problem in μ -Calculus to the word and the emptiness problem for finite automata [3]. Thus resulting an algorithm to solve the model checking and the satisfiability problem on a computer. Parity game play a crucial role within this translation, since it defines the semantics of alternating tree automata (i.e., whether an automaton accepts or rejects some transition system by the existence of a winning strategy for a player in an appropriate parity game) [8]. Parity games provide a straight forward, convent construction to complement a given alternating tree automaton and also use to show the decidability of the word problem and the emptiness problem.

This paper proceeds as in the Sec.2 transitions systems are explained along with alternating tree automaton. Sec.3 comprises of parity games and automaton acceptance. Sec.4 introduce μ -Calculus formalism where in Sec.5 we define encoding from μ -Calculus to alternating tree automaton, concluding the last section by explaining the satisfiability and model checking problems under defined encoding along with complexity bounds.

2 Transition Systems and Alternating Tree Automaton

Transition Systems are structures consisting of a non-empty set of states, a set of unary relation describing properties of states and a set of binary relation describing transitions between states. In short, transition systems are describing the operational semantics of any program. Thus model checking amounts to verifying that any corresponding transition system has a property of interest. Properties of interest are formulated in logic languages like (LTL, CTL, CTL*, PDL, μ -Calculus and MSOL);

Definition 1 (*Transition System*) A **Transition system** (i.e., Kripke Structure) over \mathcal{P} is a tripe $\mathcal{K} = (S, R, \lambda)$ where

- \mathcal{P} be a set of atomic propositions (properties) and for any propositional interpretation $\mathcal{I} : \mathcal{P} \rightarrow \{true, false\}$.
- S is a set called **states** (worlds), universe of \mathcal{K} ,
- $R \subseteq S \times S$ is a transition relation and
- $\lambda : S \rightarrow 2^{\mathcal{P}}$ is a mapping (i.e., $\lambda(s_i) = p_i$ for every $p_i \in \mathcal{P}$). $\lambda(s_i) = p_i$ if p_i is true in s_i and $\neg p_i$ if p_i is false in s_i .

$\lambda : S \rightarrow 2^{\mathcal{P}}$ regards transition systems as labeled directed graphs. For every $s \in S$, we denote

$$sR = \{s' \in S \mid (s, s') \in R\}, Rs = \{s' \in S \mid (s', s) \in R\} \quad (1)$$

A **pointed transition system** (i.e, a rooted kripke structure) is a pair (\mathcal{K}, s_I) in a transition system $\mathcal{K} = (S, R, \lambda)$ with an initial state $s_I \in S$ [3].

An alternating tree automata is a device which accepts or rejects pointed transition systems by parsing the paths.

Definition 2 (*Alternating Tree Automata*) An alternating tree is a tuple $\mathcal{A} = \{Q, q_I, \delta, \Omega\}$ where

- Q is a finite set of states of the automaton,
- $q_I \in Q$ is a state called the initial state,

- $\delta : Q \rightarrow TC^Q$ is a transition function which maps every state $q \in Q$ to a transition condition TQ where all the transition conditions TQ over Q are defined by:
 - 0 and 1 are transition conditions over Q .
 - $p, \neg p$ are transition conditions over Q , for every $p \in \mathcal{P}$.
 - $q, \Box q, \Diamond q$ are transition conditions over Q , for every $q \in Q$.
 - $q_1 \wedge q_2, q_1 \vee q_2$ are transition conditions over Q , for every $q_1, q_2 \in Q$.
- $\Omega : Q \rightarrow \omega$ is called priority function (coloring function) which assigns color to states of \mathcal{A} .

2.1 Index of Alternating Tree Automata

Concerning the complexity of an alternating tree automata, an important notion is its index [7]. Let $\mathcal{A} = \{Q, q_I, \delta, \Omega\}$ be an alternating tree automata, then the transition graph $G(\mathcal{A})$ has the set Q as vertex set. There is a edge relation from a vertex q to q' iff q' appears in the transition condition $\delta(q)$.

Let $\mathcal{C}^{\mathcal{A}}$ be the set of all strongly connected components of the transition graph $G(\mathcal{A})$ of \mathcal{A} . For every $C \in \mathcal{C}^{\mathcal{A}}$, let

$$m_C^{\mathcal{A}} = |\{\Omega^{\mathcal{A}}(q) | q \in C\}| \quad (2)$$

denote the number of priorities used in C . The index of \mathcal{A} , denoted by $index(\mathcal{A})$, is the maximum of all these values, that is,

$$index(\mathcal{A}) = \max(\{m_C^{\mathcal{A}} | C \in \mathcal{C}^{\mathcal{A}}\} \cup \{0\}) \quad (3)$$

3 Infinite Games & Parity Acceptance Conditions

This section introduce infinite two-person games on directed graphs along with winning play strategies for a certain player [5].

Definition 3 (Game)

A game is composed of an arena and a winding condition. Let \mathcal{A} be an arena then the pair $\mathcal{G} = (\mathcal{A}, Win)$ is called a **game** where $Win \subseteq V^\omega$ is a winning set where ω is infinite supply of intergers.

Definition 4 (Arena) An arena is a triple

$$\mathcal{A} = (V_0, V_1, E) \quad (4)$$

where V_0 is a set of 0-vertices, V_1 a set of 1-vertices and $E \subseteq (V_0 \cup V_1) \times (V_0 \cup V_1)$ is the **edge relation** also known as set of moves. The union $V = (V_0 \cup V_1)$ where V_0 and V_1 are disjoint sets of vertices. Under this union requirement edge relation correspond to $E \subseteq V \times V$. The set of **successor** of $v \in V$ is defined as $vE = \{v' \in V | (v, v') \in E\}$. Consider a player σ for $\sigma \in \{0, 1\}$, then the opponent of player σ is player $\bar{\sigma}$ (i.e., $\bar{\sigma} = 1 - \sigma$).

Definition 5 (Play) We define a play in the arena \mathcal{A} as followed:

- a finite play $\pi = v_0 v_1 \dots v_l \in V^+$ with $v_{i+1} \in v_i E$ for all $i < l$ and $v_l E = \emptyset$ represents a dead-end, a prefix of this finite play is $\rho(\pi) = v_0 v_1 \dots v_k$ for $k \leq l$.

- an infinite play $\pi = v_0 v_1 \dots v_l \in V^\omega$ with $v_{i+1} \in v_i E$ for all $i \in \omega$, a prefix for this infinite play is $\rho(\pi) = v_0 v_1 \dots v_k$ for $k \geq 0$

Definition 6 (*Winning Set*) To define the winning conditions for Players (Player 0, Player 1) are as followed:

Player σ is declared the winner of a play π in the game \mathcal{G} iff

- π is a finite play $\pi = v_0 v_1 \dots v_l \in V^+$ and v_l is a $\bar{\sigma}$ -Vertex where Player $\bar{\sigma}$ can not move anymore (i.e., v_l is a dead-end, $v_l E = \emptyset$) or
- π is an infinite play and $\pi \in Win$.

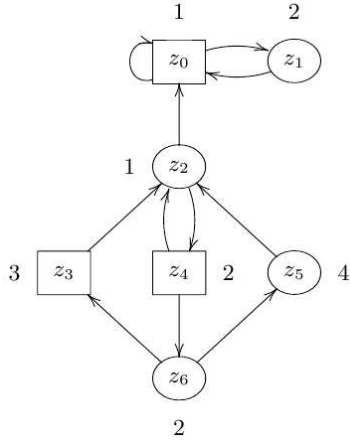
Conversely, Player $\bar{\sigma}$ wins play π if Player σ does not win π .

In every play, a token is placed on some initial vertex $v \in V$. If v is σ -vertex then Player σ moves the token from v to $v' \in vE$, symmetrically moves for Player $\bar{\sigma}$ are considered in case of $\bar{\sigma}$ -vertex. This process is repeated infinity often or until a dead end is reached (i.e., a vertex without successor).

Definition 7 (*Coloring Function*) The coloring function $\chi : V \rightarrow C$ color vertices of arena \mathcal{A} where C is a finite set of colors (priorities)(i.e, $C \subseteq \mathbb{N}$) and it extends to an infinite play $\pi = v_0 v_1 \dots$ as $\chi(\pi) = \chi(v_0)\chi(v_1)\dots$.

Let Win is an acceptance condition for an automaton then $W_\chi(Win)$ is the winning set consisting of all infinite plays π where $\chi(\pi)$ is accepted according to Win .

- Parity conditions or Colour set C is a finite subset of integers and $Inf(\chi(\pi))$ be the set of colors that occurs infinitely often in $\chi(\pi)$ then for,
 Max-parity condition: $\pi \in W_\chi(Win)$ iff $\max(Inf(\chi(\pi)))$ is even.
 Min-parity condition: $\pi \in W_\chi(Win)$ iff $\min(Inf(\chi(\pi)))$ is even.



Example 1

In above figure, let $(G) = (\mathcal{A}, Win)$ defines an arena where $\mathcal{A} = (V_0, V_1, E)$ such that $V_0 = \{z_1, z_2, z_5, z_6\}$ (circles), $V_1 = \{z_0, z_3, z_4\}$ (squares), Coloring set $C = \{1, 2, 3, 4\}$ and $\chi(z_4) = 2$ as shows in figure; winning set of condition $Win = \{\{1, 2\}, \{1, 2, 3, 4\}\}$. In a possible infinite play in this is $\pi = z_6 z_3 z_2 z_4 z_2 z_4 z_6 z_5 (z_2 z_4)^\omega$. According to Muller acceptance condition (i.e., $\pi \in W_\chi(Win)$ iff $Inf(\chi(\pi)) \in \mathcal{A}$) this play π is winning for Player 0 because $\chi(\pi) = 23121224(12)^\omega$ where $Inf(\chi(\pi)) = \{1, 2\} \in Win$. For play $\pi' = (z_2 z_4 z_6 z_3)^\omega$ yields $\chi(\pi') = (1223)^\omega$ and $Inf(\chi(\pi')) =$

$\{1, 2, 3\} \notin \text{Win}$, hence π' is winning for Player 1. Regarding parity conditions this play is a loss for player 0 because $\min(\text{Inf}(\chi(\pi)) = \{1\})$ is odd., hence a win for the opponent.

3.1 Behavior of Alternating Tree Automata

Let (\mathcal{K}, s_I) be a pointed transition system and let $\mathcal{A} = (Q, q_I, \delta, \Omega)$ be an alternating tree automaton. To define the behavior of \mathcal{A} on (\mathcal{K}, s_I) consider alphabet over sequences of pairs $(Q \times S)$ [3].

For a word $w \in (Q \times S)^*$ where $(q, s) \in (Q \times S)$ is an alphabet such that the *behavior* of \mathcal{A} on (\mathcal{K}, s_I) is the language $\mathcal{L}(\mathcal{A}, \mathcal{K}) \subseteq (Q \times S)^*$ (i.e., $w \in L(\mathcal{A}, \mathcal{K})$). The initial state correspond to $(q_I, s_I) \in \mathcal{L}(\mathcal{A}, \mathcal{K})$. To define the transition function consider that the automaton is in the state q and it inspects the state s (i.e., the current instance is (q, s)). Now, the automaton tries to execute the transition condition $\delta(q)$ defined as followed:

- If $\delta(q) \in \{0, 1\}$ or $\delta(q) = p$ or $\delta(q) = \neg p$ for some proposition $p \in \mathcal{P}$ then the automaton \mathcal{A} need not to take any action.
- If $\delta(q) = q'$ for some $q' \in Q$, then from current state updates $(q, s) \rightarrow (q', s)$, then $(q, s)(q', s) \in \mathcal{L}(\mathcal{A}, \mathcal{K})$.
- If $\delta(q) = q_1 \wedge q_2$ or $\delta(q) = q_1 \vee q_2$ for some $q \in Q$, then \mathcal{A} splits itself into two instances (q_1, s) and (q_2, s) such that $(q_1, s)(q_2, s) \in \mathcal{L}(\mathcal{A}, \mathcal{K})$.
- If $\delta(q) = \Box q'$ or $\delta(q) = \Diamond q'$ for some $q' \in Q$, then automaton splits in several instances such that for every $s' \in sR$, $(q, s)(q', s') \in \mathcal{L}(\mathcal{A}, \mathcal{K})$.

Now the main concern is how the automaton \mathcal{A} *accepts or rejects* the pointed transition system (\mathcal{K}, s_I) .

Lets developed a notion of a **successful instance** for an instance (q, s) such that:

- If $\delta(q) = 1$, then the instance succeeds and if $\delta(q) = 0$, then it does not succeed.
- If $\delta(q)$ is a propositional variable $p \in \mathcal{P}$ and p is true in the state s (i.e., $\delta(q) = p$ and $p \in \lambda(s)$) then instance is successful. Similarly, holds for $\delta(q) = \neg p$ and $p \notin \lambda(s)$.
- Converseley, if $\delta(q) = p$ and $p \notin \lambda(s)$ or $\delta(q) = \neg p$ and $p \in \lambda(s)$ then the instance (q, s) is not successful.
- If $\delta(q) = q'$, then automaton changes its state to new instance (q', s) such that the instance (q, s) is successful iff (q', s) is successful.
- If $\delta(q_1 \wedge q_2)$, then the instance (q, s) succeeds iff both instances (q_1, s) and (q_2, s) succeed.
- If $\delta(q_1 \vee q_2)$, then the instance (q, s) succeeds iff at least one of the the instance (q_1, s) and (q_2, s) succeeds.
- If $\delta(q) = \Box q'$, then iinstance (q, s) succeeds iff $\forall s' \in sR$ the instances (q, s') succeeds.
- Finally, if $\delta(q) = \Diamond q'$, then instance (q, s) succeeds iff $\exists s' \in sR$ such that an instance (q, s') succeeds.

The result of this process is a parse tree with instances as nodes;

Definition 8 The automaton \mathcal{A} accepts the finite transition system (\mathcal{K}, s_I) iff the initial instance $(q_I, s_I) \in \mathcal{L}(\mathcal{A}, \mathcal{K})$ succeeds.

Above mentioned formalization for the notion of *successful instance* encounter following problems:

- If parse tree is infinite, then successful instances cannot be determined in a bottom-up fashion.
- If $\delta(q) = q'$, then the instance (q, s) is successful iff (q', s) is successful. However, if $\delta(q) = q$, then it end up in an *infinite loop*.

To resolve these problems; "evaluation problem" is considered as solving a certain game in an infinite play where acceptance is decided according to a winning condition as explained in the next section of parity games.

3.2 Automata acceptance using Parity conditions

The automaton $\mathcal{B} = (Q, q_I, \delta, \Omega)$ **accepts** the pointed transition system (\mathcal{K}, s_I) iff there a winning strategy for Player 0 in the \mathcal{G} where $\mathcal{G} = (\mathcal{A}, Win)$ is a two player (i.e., Player 0, Player 1) parity game where $\mathcal{A} = (V_0, V_1, E)$ defines an arena, $V = (V_0 \cup V_1)$ and $Win \subseteq V^\omega$ is a winning set such that:

A vertex $v = (q, s) \in V_0$ where $q \in Q$ and $s \in S$ iff

- $\delta(q) = 0$.
- $\delta(q) = p, p \notin \lambda(s)$.
- $\delta(q) = \neg p, p \in \lambda(s)$.
- $\delta(q) = q'$ for some $q' \in Q$.
- $\delta(q) = q_1 \vee q_2$ for $q_1, q_2 \in Q$
- $\delta(q) = \Diamond q'$.

A vertex $v = (q, s) \in V_1$ iff

- $\delta(q) = 1$.
- $\delta(q) = p, p \in \lambda(s)$.
- $\delta(q) = \neg p, p \notin \lambda(s)$.
- $\delta(q) = q_1 \wedge q_2$ for $q_1, q_2 \in Q$
- $\delta(q) = \Box q'$.

In parity Game \mathcal{G} , the edge relation E is defined as:

$$E := \{((q, s), (q_1, s_1)) | (q_1, s_1) \in (q, s)R \wedge (q, s), (q_1, s_1) \in V\}. \quad (5)$$

For alternating tree automata \mathcal{B} the priority function is defined by:

$$\Omega(q, s) := \Omega(q) \text{ for } (q, s) \in V. \quad (6)$$

3.3 Winning Strategy

A strategy for a Player σ is the function $f_\sigma : V_\sigma \times V \rightarrow V$ and a prefix play for $\pi = v_0 v_1 \dots v_l$ is conform with f_σ if for every i with $0 \leq i < l$ and $v_i \in V_\sigma$ is defined and we have for $v_{i+1} = f_\sigma(v_0 \dots v_i)$. A play is conform with f_σ if each of its prefix is conform with f_σ . f_σ is a strategy for Player σ on $U \subseteq V$ if it is defined for every prefix of a play which is conform with it, starts in a vertex U and does not end in a dead end for player σ . A strategy f_σ is a winning strategy for Player σ on U if all plays which are conform with f_σ and start from a vertex in U are wins for $Player\sigma$.

Definition 9 Player σ wins a game \mathcal{G} on $U \subseteq V$ if he has a winning strategy on U .

The winning region for Player σ is the set $W_\sigma(\mathcal{G}) \subseteq V$ of all vertices such that Player σ wins (\mathcal{G}, v) (i.e., Player 0 wins \mathcal{G} on $\{v\}$). Hence, for any \mathcal{G} , Player σ wins \mathcal{G} on $W_\sigma(\mathcal{G})$.

Definition 10 The automaton $\mathcal{A} = \{Q, q_I, \delta, \Omega\}$ accepts the pointed transition system (\mathcal{K}, s_I) iff there is a winning strategy for Player 0 in the parity game $\mathcal{G} = ((V_0, V_1, E), \Omega(q))$.

The language of automaton \mathcal{A} consists of the pointed transition systems which \mathcal{A} accepts and is denoted by $L(\mathcal{A})$ [3].

Example 2 Let $\delta(q) = \Box q_I$ and $\Omega(q_I) = 0$. Let (\mathcal{K}, s_I) be any pointed transition system. Player 1 can't win since he loses every finite and infinite play because the only priority function $\Omega(q) = 0$, which complies to single vertex $(q_I, s_I) \in V_0$ with infinite loop itself hence accepts every pointed transition system.

Example 3 Let $\delta(q) = \Diamond q_I$ and $\Omega(q_I) = 1$. This automaton does not accept any pointed transition system.

Proposition 1 (*Word Problem*) The word problem is to decide whether a given alternating tree automaton $\mathcal{A} = \{Q, q_I, \delta, \Omega\}$ accepts a given finite pointed transition system (\mathcal{K}, s_I) .

Proposition 2 (*Emptiness Problem*) The emptiness problem is to show that an alternating tree automaton $\mathcal{A} = \{Q, q_I, \delta, \Omega\}$ accepts if \mathcal{A} accepts at least one transition system.

4 μ -Calculus

4.1 Syntax and Semantics of μ -Calculus

This section introduces modal μ -Calculus by presenting syntax and semantics and later we introduce the notion of Tarski Fixed point theorem;

4.2 Syntax of μ -Calculus

Definition 11 The set L_μ is a set of inductively defined modal μ -Calculus formulas:

- $\perp, \top \in L_\mu$.
- For every atomic proposition $p \in \mathcal{P}$; $p, \neg p \in L_\mu$.
- If $\varphi, \psi \in L_\mu$, then $\varphi \circ \psi \in L_\mu$ where $\circ \in \{\vee, \wedge\}$.
- If $\varphi \in L_\mu$, then $\Box \varphi, \Diamond \varphi \in L_\mu$.
- If $p \in \mathcal{P}$, $\varphi \in L_\mu$, and p occurs only positively in φ then $\mu p \varphi, \nu p \varphi \in L_\mu$.

4.3 Fixed point Operators & Free variable set

The operators μ and ν are called *fixed point operators* (i.e., μ , least fixed-point and ν , greatest fixed-point) viewed as quantifiers ensuring that the argument of a fixed-point operator as a monotone function [8]. Accordingly, the set $free(\varphi)$ of free variables of an L_μ formula φ is defined inductively as follows:

- $free(\perp) = free(\top) = \emptyset$,
- $free(p) = free(\neg p) = \{p\}$,
- $free(\varphi \circ \psi) = free(\varphi) \cup free(\psi)$ where $\circ \in \{\wedge, \vee\}$,
- $free(\Box\varphi) = free(\Diamond\varphi) = free(\varphi)$,
- $free(\mu p\varphi) = free(\nu p\varphi) = free(\varphi)/\{p\}$.

The sets F_μ and F_ν are defined as follows:

$$F_\mu = \{\mu p\psi \mid \psi \in L_\mu\}, F_\nu = \{\nu p\psi \mid \psi \in L_\nu\}. \quad (7)$$

Formulas from the set $F_\eta = F_\mu \cup F_\nu$ are called fixed point formulas.

4.4 Fixed point Alternation

Beside its length, the most important characteristic of a formula $\varphi \in L_\mu$ is its fixed point alternation depth, that is, the number of alternations between least and greatest fixed point operators. We now define the notion of alternation depth for L_μ formulas which coincides with the notion of index for an alternating tree automaton (i.e., $\alpha(\varphi) = index(\mathcal{A}(\varphi))$) [7].

Definition 12 (*Syntactic alternation depth*) For an arbitrary formula $\varphi \in L_\mu$, its alternation depth $\alpha(\varphi) : L_\mu \rightarrow \mathbb{N}$ is function defined inductively:

- $\alpha(\perp) = \alpha(\top) = \alpha(p) = \alpha(\neg p) = 0$,
- $\alpha(\varphi \circ \psi) = \max\{\alpha(\varphi), \alpha(\psi)\}$ where $\circ \in \{\wedge, \vee\}$,
- $\alpha(\Box\varphi) = \alpha(\Diamond\varphi) = \alpha(\varphi)$,
- $\alpha(\mu p\varphi) = \max\{1, \alpha(\psi)\} \cup \{\alpha(\nu p'\psi') + 1 \mid \nu p'\psi' \leq \psi, p \in free(\nu p'\psi')\}$,
- $\alpha(\nu p\varphi) = \max\{1, \alpha(\psi)\} \cup \{\alpha(\mu p'\psi') + 1 \mid \mu p'\psi' \leq \psi, p \in free(\mu p'\psi')\}$.

Alternation depth of a formula is greater or equal to the alternation depth of any subformula. Note that, computing the alternation depth of all subformulae of an L_μ can be done in $\mathcal{O}((|\varphi|_{\mathcal{K}})^2 + |\varphi| \log |\varphi|) \subseteq \mathcal{O}(|\varphi|^2)$ time in bottom up fashion.

Example 4 Let consider a formula $\varphi = \mu p_1(\nu p_1(p_0 \wedge p_1) \vee \Diamond p_1)$, its alternation depth is $\alpha(\varphi) = 1$ and $\psi = \mu p_1((p_2 \wedge p_0) \vee p_1)$ and $\varphi = \nu p_2(\psi)$ it follows $\alpha(\varphi) = 2$ since $p_1 \in free(\psi)$.

4.5 Lattice Theory & Monotone Functions

The semantics of the μ -Calculus is anchored in the Tarski-Knaster theorem, giving a means to do iteration based model checking in an efficient manner. In order to introduce the concept of Tarski theorem some basic notions of lattice theory and fixed-point are revisited as followed:

Definition 13 (*Lattice Theory*) A lattice (L, \leq) consists of a set L and a partial order \leq such that any pair of elements has greatest lower bound, the meet \sqcap , and a least upper bound, the join \sqcup , with following properties:

$$(associative\ law)(x \sqcup y) \sqcup z = x \sqcup (y \sqcup z). \quad (8)$$

$$(x \sqcap y) \sqcap z = x \sqcap (y \sqcap z). \quad (9)$$

$$(commutative\ law)(x \sqcup y) = (y \sqcup x). \quad (10)$$

$$(x \sqcap y) = (y \sqcap x). \quad (11)$$

$$(idempotency\ law)(x \sqcup x) = (x \sqcup x). \quad (12)$$

$$(x \sqcap x) = (x \sqcap x). \quad (13)$$

$$(absorption\ law)x \sqcup (x \sqcap y) = x. \quad (14)$$

$$x \sqcap (x \sqcup y) = x. \quad (15)$$

Example 5 Given a set S , the power set of S , (i.e., $\mathcal{P}(S)$) is $(\mathcal{P}(S), \subseteq)$ is a lattice.

A Lattice $(L, \leq, \sqcup, \sqcap)$ is *complete* if $\forall A \subseteq L$ implies $\sqcup A$ and $\sqcap A$ are defined and there also exists a *minimum element* (i.e., $\perp = \sqcap L$) and a *maximum element* (i.e., $\top = \sqcup L$).

Example 6 Given a set S , the power set of S , (i.e., $\mathcal{P}(S)$) is $(\mathcal{P}(S), \subseteq)$ is a lattice. For a given set $A \subseteq \mathcal{P}(S)$ of subsets such that maximal set $\sqcup A = \bigcup_{S' \in A} S'$ and minimal set $\sqcap A = \bigcap_{S' \in A} S'$:

$$\sqcup A = \left\{ \bigcup_{S' \in A} S' \mid A \subseteq \mathcal{P}(S) \right\} \quad (16)$$

$$\sqcap A = \left\{ \bigcap_{S' \in A} S' \mid A \subseteq \mathcal{P}(S) \right\} \quad (17)$$

$$(18)$$

Definition 14 (*Monotone Functions*)

A monotonic function (or monotone function) is a function which preserves the given order is formalized as followed:

- $f : L \rightarrow L$ is a monotonic order preserving if

$$\forall x, y \in L. x \leq y \Rightarrow f(x) \leq f(y). \quad (19)$$

- x is a fix-point if $f(x) = x$.

f^0 is an identity function and $f^{n+1} = f^n \circ f^0$, f monotonic implies that f^n is also monotonic. The identity function is monotonic and composing two monotonic functions gives a monotonic function.

4.6 Tarski-Knaster Fix-point Theorem

Theorem 1 Let $f : L \rightarrow L$ be a monotonic function on a complete lattice $(L, \leq, \sqcup, \sqcap)$ then for $A = \{y \mid f(y) \leq y\}$, $x = \sqcap A$ is the least fixed point of f .

Proof sketch:

- (1) $f(x) \leq x$: $\forall y \in A, x \leq y$ therefore $f(x) \leq f(y) \leq (x)$. So, $f(x) = \sqcap A = x$.
- (2) $x \leq f(x)$: by monotonicity applied to 1, $f^2(x) \leq f(y)$ so $f(x) \in A$, and $x \leq f(x)$. Thus x is a fixed point and because all fixed points belong to A , x is the least fixed point. Similarly for the greatest fix-point with $A = \{y | f(y) \geq y\}$.

4.7 Semantics of μ -Calculus

The formulas of modal μ -Calculus are interpreted in Kripke structures \mathcal{K} such that for every Kripke structure \mathcal{K} and every $\varphi \in L_\mu$, where $\kappa := \mathcal{P} \rightarrow 2^S$ is defined as:

- $\|\perp\|_{\mathcal{K}} = \emptyset, \|\top\|_{\mathcal{K}} = S,$
- $\|p\|_{\mathcal{K}} = \kappa(p), \|\neg p\|_{\mathcal{K}} = S/\kappa(p),$
- $\|\varphi_1 \vee \varphi_2\|_{\mathcal{K}} = \|\varphi_1\|_{\mathcal{K}} \cup \|\varphi_2\|_{\mathcal{K}},$
- $\|\varphi_1 \wedge \varphi_2\|_{\mathcal{K}} = \|\varphi_1\|_{\mathcal{K}} \cap \|\varphi_2\|_{\mathcal{K}},$
- $\|\Box\varphi\|_{\mathcal{K}} = \{s \in S | sR \subseteq \|\varphi\|_{\mathcal{K}}\},$
- $\|\Diamond\varphi\|_{\mathcal{K}} = \{s \in S | sR \cap \|\varphi\|_{\mathcal{K}} \neq \emptyset\}.$

We will say that pointed Kripke structure (\mathcal{K}, s_I) is a model of $\varphi \in L_\mu$, denoted by $\mathcal{K} \models \varphi$ if $s_I \in \|\varphi\|_{\mathcal{K}}$. Additionally, we write $\varphi \equiv \psi$ if for all Kripke models (\mathcal{K}, s_I) , we have $(\mathcal{K}, s_I) \models \varphi$ iff $(\mathcal{K}, s_I) \models \psi$.

To define the semantics of the fixed-point operators where \mathcal{K} is a Kripke structure, p is a propositional variable and $S' \subseteq S$ then $\mathcal{K}[p \mapsto S']$ denotes the Kripke structure as followed:

$$\mathcal{K}[p \mapsto S'] = (S, E, \kappa[p \mapsto S']) \quad (20)$$

where $\kappa[p \mapsto S']$ is given as followed:

$$\kappa[p \mapsto S']_{p'} = \begin{cases} S' & \text{if } p' = p \\ \kappa(p) & \text{if } p' \neq p \end{cases}$$

The semantics of the fixed-point operators is now defined as:

- $\|\mu p \varphi\|_{\mathcal{K}} = \bigcap \{S' \subseteq S | \|\varphi\|_{\mathcal{K}[p \mapsto S']} \subseteq S'\}$
- $\|\nu p \varphi\|_{\mathcal{K}} = \bigcup \{S' \subseteq S | \|\varphi\|_{\mathcal{K}[p \mapsto S']} \supseteq S'\}$

$\mu z.f(z)$, the least fix-point of f is equal to $\sqcup_i f^i(\emptyset)$, where i ranges over all ordinals of cardinality at most the state space L ; when L is finite, $\mu z.f(z)$ is the union of following ascending chain $\emptyset \subseteq f(\emptyset) \subseteq f^2(\emptyset) \dots$

$\nu z.f(z) = \sqcap_i f^i(\top)$, where i ranges over all ordinals of cardinality at most the state space L ; when L is finite, $\nu z.f(z)$ is the intersection of following descending chain $\top \supseteq f(\top) \supseteq f^2(\top) \dots$

Example 7 Consider a formula $\varphi = \mu p(\Box p)$. This formula characterizes those worlds of a Kripke model where only path with finite length exists. That is, we have $s_I \in \|\varphi\|_{\mathcal{K}}$ iff all paths in \mathcal{K} starting in s_I are finite. Lets consider this formula evaluation in detail;

Proof " \Leftarrow ": If all paths in \mathcal{K} starting in s_I are finite then $s_I \in \|\varphi\|_{\mathcal{K}}$ if all paths in \mathcal{K} starting in s_I are finite.

Let consider $\mathcal{K} = (S, R, \lambda)$ be an arbitrary Kripke model. In order to show $S_f \subseteq \|\varphi\|_{\mathcal{K}}$, Suppose $S_f \subseteq S$ denote only states(worlds) of finite length paths. Accordingly to above defined semantics of L_μ for least fixed point $\|\Box\varphi\|_{\mathcal{K}[p \mapsto S']}$ implies $S_f \subseteq S$ holds for all $S' \subseteq S$. By unwinding \Box operator $\|\Box\varphi\|_{\mathcal{K}[p \mapsto S']} = \{s \in S | sR \subseteq \|\varphi\|_{\mathcal{K}[p \mapsto S']}\}$ implies $S_f \subseteq S'$ (i.e., All paths are finite).

5 Encoding μ -Calculus to Alternating Tree Automaton ($L_\mu = L_{\mathcal{A}}$)

To define the translation of μ -Calculus formulas into an alternating tree automaton \mathcal{A} , let us consider a notion of subformula property and a mapping for every subformula to a state tuple [8] [7].

Definition 15 For every formula $\varphi \in L_\mu$, subformula of φ defined as follows:

- φ is subformula of $\varphi \in L_\mu$,
- φ, ψ is subformula of $\varphi \circ \psi \in L_\mu$ where $\circ \in \{\wedge, \vee\}$,
- φ is subformula of $\Box\varphi, \Diamond\varphi, \mu p\varphi, \nu p\varphi \in L_\mu$.

The function which simply unwinds the formula by defining a state correspondence for every consecutive subformula:

$$||\varphi|| = \langle \varphi_0, \dots, \varphi_n \rangle \text{ for every } \varphi_i \in \text{Sub}(\varphi), i \leq |\varphi|. \quad (21)$$

Let φ be an L_μ formula in normal form. The alternating tree automaton $\mathcal{A}(\varphi)$ is defined by:

$$\mathcal{A} = (Q, q_I, \delta, \Omega). \quad (22)$$

where

- Q is the set which contains for each subformula ψ of φ (including φ itself), a state denoted by $\langle \psi \rangle$,
- the initial state is given by $q_I = \langle \varphi \rangle$.

The transition relation δ is defined by:

- $\delta(\langle \perp \rangle) = 0$,
- $\delta(\langle \top \rangle) = 1$,
- $\delta(\langle p \rangle) = \begin{cases} p & \text{if } p \in \text{free}(\varphi) \\ \langle \varphi_p \rangle & \text{if } p \notin \text{free}(\varphi) \end{cases}$
- $\delta(\langle \neg p \rangle) = \neg p$,
- $\delta(\langle \psi_1 \wedge \psi_2 \rangle) = \langle \psi_1 \rangle \wedge \langle \psi_2 \rangle$, $\delta(\langle \psi_1 \vee \psi_2 \rangle) = \langle \psi_1 \rangle \vee \langle \psi_2 \rangle$,
- $\delta(\langle \Diamond \psi \rangle) = \Diamond \langle \psi \rangle$,
- $\delta(\langle \Box \psi \rangle) = \Box \langle \psi \rangle$,
- $\delta(\langle \mu p \psi \rangle) = \langle \psi \rangle$,
- $\delta(\langle \nu p \psi \rangle) = \langle \psi \rangle$.

The priority function $\Omega : Q \rightarrow \omega$ is defined by:

$$\Omega(\langle \psi \rangle) = \begin{cases} 2\lceil \alpha(\psi)/2 \rceil - 1 & \text{if } \psi \in F_\mu, \alpha(\psi) > 0 \\ 2\lfloor \alpha(\psi)/2 \rfloor & \text{if } \psi \in F_\nu, \alpha(\psi) > 0 \\ 0 & \text{otherwise} \end{cases}$$

The notion of index of an alternating tree automaton coincide with the notion of alternation depth. Let φ be an L_μ formula then $\alpha(\varphi) = \text{index}(\mathcal{A}(\varphi))$.

Example 8 Consider $\varphi = \mu q_0(q_0 \vee q_1)$ then the transition function with defined mapping is as follows:

$$\delta(\langle \mu q_0(q_0 \vee q_1) \rangle) = \langle q_0 \vee q_1 \rangle \quad (23)$$

$$\delta(\langle q_0 \vee q_1 \rangle) = \langle q_0 \rangle \vee \langle q_1 \rangle \quad (24)$$

$$\delta(\langle q_0 \rangle) = q_0. \quad (25)$$

A vertex $v = (\langle \psi \rangle, s)$ belongs to Player 0 iff

- $\psi = \perp$,
- $\psi = p, p \in \text{free}(\varphi), s \notin \lambda(s)$,
- $\psi = \neg p, p \in \text{free}(\varphi), s \in \lambda(s)$,
- $\psi = p, p \notin \text{free}(\varphi)$,
- $\psi = \eta p \psi'$ where $\eta \in \{\mu, \nu\}$,
- $\psi = \psi_1 \vee \psi_2$ for some $\psi_1, \psi_2 \in L_\mu$,
- $\psi = \Diamond \psi'$.

A vertex $v = (\langle \psi \rangle, s)$ belongs to Player 1 iff

- $\psi = \top$,
- $\psi = p, p \in \text{free}(\varphi), s \in \lambda(s)$,
- $\psi = \neg p, p \in \text{free}(\varphi), s \notin \lambda(s)$,
- $\psi = p, p \in \text{free}(\varphi)$,
- $\psi = \psi_1 \wedge \psi_2$ for some $\psi_1, \psi_2 \in L_\mu$,
- $\psi = \Box \psi'$.

In parity Game \mathcal{G} , the edge relation $E^\mathcal{G}$ is defined as:

$$E^\mathcal{G} = \begin{cases} \{(\langle \psi' \rangle, s) | \langle \psi' \rangle \in \delta(\langle \psi \rangle)\} & \text{if } \psi \neq \Diamond \psi', \Box \psi' \\ \{(\langle \psi' \rangle, s') | \langle \psi' \rangle \in \delta(\langle \psi \rangle), s' \in sR\} & \text{if } \psi = \Diamond \psi', \Box \psi' \end{cases}$$

For μ -formula the priority is odd and for a ν -formula priority is even.

Theorem 2 Let φ be an arbitrary L_μ formula. Then φ and $\mathcal{A}(\varphi)$ are equivalent, that is:

$$\|\varphi\| = \|\mathcal{A}(\varphi)\|. \quad (26)$$

6 Proof of Correctness

Theorem 3 Let φ be an arbitrary L_μ formula. Then for every pointed transition system (\mathcal{K}, s) the following holds:

$$(\mathcal{K}, s) \models \varphi \text{ iff } (\mathcal{K}, s) \in L(\mathcal{A}(\varphi)) \quad (27)$$

Proof We proceed by induction on the size of the formula φ :

Case: $\varphi = \top$. Clearly, every Kripke structure (\mathcal{K}, s_I) is a model of φ . Thus every pointed transition system is accepted by $\mathcal{A}(\varphi)$. The initial state of game $\mathcal{G}(\mathcal{A}(\varphi), \mathcal{K}, s_I)$ is a *Vertex* – 1 and is dead-end. Hence, every game in this play is won by Player 0.

Case: $\varphi = \perp$. Then for the complement case $(\mathcal{K}, s) \not\models \perp$ and from proposition 1 it follows that automata \mathcal{A} does not contain any succeeding run for $\varphi = \perp$ (i.e., $(\mathcal{K}, s) \notin L(\mathcal{A}(\varphi))$).

Case: $\varphi = p$. Let $(\mathcal{K}, s_I) \models \varphi$ if $s_I \in \kappa(p)$. Thus in $\mathcal{G}(\mathcal{A}(\varphi), \mathcal{K}, s_I)$ is *vertex* – 1 and a deadend as well, therefore $(\mathcal{K}, s_I) \in L(\mathcal{A}(\varphi))$. Similarly, if $(\mathcal{K}, s_I) \not\models \varphi$ if $\kappa(p) \in s_I$ then we have $s_I \notin \kappa(p)$, thus $\mathcal{G}(\mathcal{A}(\varphi), \mathcal{K}, s_I)$ is *vertex* – 0 and a dead-end. Therefore, $(\mathcal{K}, s_I) \notin L(\mathcal{A}(\varphi))$.

Case: $\varphi = \neg p$. Similar to the previous case.

Case: $\varphi = \psi_1 \wedge \psi_2$, then:

$$(\mathcal{K}, s_I) \models \psi_1 \wedge \psi_2 = (\mathcal{K}, s_I) \models \psi_1 \wedge \psi_2 \quad (28)$$

$$= s_I \in \|\psi_1\|_{\mathcal{K}} \cap s_I \in \|\psi_2\|_{\mathcal{K}} \quad (L_{\mu}, \text{induction}) \quad (29)$$

$$= (\mathcal{K}, s_I) \models \psi_1 \text{ and } (\mathcal{K}, s_I) \models \psi_2 \quad (\text{def.}) \quad (30)$$

$$= (\mathcal{K}, s_I) \in L(\mathcal{A}(\psi_1)) \cap L(\mathcal{A}(\psi_2)) \quad (\text{lemma.3}^1) \quad (31)$$

$$= (\mathcal{K}, s_I) \in L(\mathcal{A}(\psi_1 \wedge \psi_2)) \quad (\text{lemma.3}^1) \quad (32)$$

Case $\varphi = \psi_1 \vee \psi_2$, similar to the previous case, *lemma.4*¹ is used instead.

Case $\varphi = \Box\psi$

$$(\mathcal{K}, s_I) \models \Box\varphi = s_I \in \|\Box\psi\|_{\mathcal{K}} \quad (\text{definition}) \quad (33)$$

$$= s_I \in \|\psi\|_{\mathcal{K}} \quad (L_{\mu}) \quad (34)$$

$$= \forall s' \in sR. (\mathcal{K}, s') \models \psi. \quad (\text{def.}) \quad (35)$$

$$= \forall s' \in sR. (\mathcal{K}, s') \in L(\mathcal{A}(\psi)). \quad (\text{def.}) \quad (36)$$

$$= (\mathcal{K}, s') \in L(\mathcal{A}(\Box\psi)). \quad (\text{lemma.5}^1) \quad (37)$$

Case $\varphi = \Diamond\psi$, similar to the previous case, *lemma.6*¹ is used instead.

Case $\varphi = \mu p\psi$, Let (\mathcal{K}, s_I) be a pointed Kripke structure where $\mathcal{K} = (S, R, \lambda)$, consider a monotone function $g : 2^S \rightarrow 2^S$ such that:

$$g(S') = \{s' \in S \mid \forall S' \subseteq S. (\mathcal{K}[p \mapsto S'], s') \in L(\mathcal{A}(\psi))\} \quad (38)$$

$$(\mathcal{K}, s) \in L(\mathcal{A}(\mu p\psi)) \text{ iff Player 0 wins the game } \mathcal{G} = (\mathcal{A}(\mu p\psi), \mathcal{K}, s). \quad (39)$$

Two further notions μg , a montone function and S_{μ} , set of winning positions of Player 0 are defined as:

$$\mu g = \{S' \subseteq S \mid g(S') \subseteq S'\} \quad (40)$$

$$S_{\mu} = \{s \in S' \mid \text{There is a winning strategy } f \text{ for Player 0 in } \mathcal{G}(\mathcal{A}(\mu p\psi), \mathcal{K}, s)\}. \quad (41)$$

Hence we have to show that $\mu g = S_{\mu}$

" \subseteq :" It's suffice to show that $g(S_{\mu}) \subseteq S_{\mu}$. Let $s \in g(S_{\mu})$, Player 0 has a memoryless winning strategy f in game $\mathcal{G}(\mathcal{A}(\mu p\psi), \mathcal{K}[p \mapsto S_{\mu}], s)$. The $\mathcal{G}(\mathcal{A}(\mu p\psi), \mathcal{K}, s)$ has an initial vertex $(\langle \mu p\psi \rangle, s)$ that has an edge to vertex $(\langle \psi \rangle, s)$ in game $\mathcal{G}(\mathcal{A}(\mu p\psi), \mathcal{K}[p \mapsto S_{\mu}], s)$. At first Player 0 moves the pebble to $(\langle \psi \rangle, s)$ then by playing in accordance with his memoryless winning strategy f for the game $\mathcal{G}(\mathcal{A}(\mu p\psi), \mathcal{K}[p \mapsto S_{\mu}], s)$ the play reaches a vertex of form $(\langle p \rangle, s)$ which is a dead end in this game. Since Player 0 has played with f , this vertex must belonged to Player 1, that is $s \in S_{\mu} = \kappa[p \mapsto S_{\mu}](p)$. Thus by definition of S_{μ} , Player 0 has a winning strategy in $\mathcal{G}(\mathcal{A}(\mu p\psi), \mathcal{K}[p \mapsto S_{\mu}], s)$ since he move the pebble to $(\langle \mu p\psi \rangle, s)$ and then play in accordance with his winning strategy and wins. Therefore $g(S_{\mu}) \subseteq S_{\mu}$.

¹ Appendix for listed proof;

" \supseteq :" In order to prove that $S_\mu \subseteq S'$, now proceed as proof by contradiction. Let $s_1 \in S_\mu$ but $s_1 \notin S'$. Since f is a winning strategy for Player 0, so restriction of f to the vertices of $\mathcal{G}(\mathcal{A}(\psi), \mathcal{K}[p \mapsto S'], s_1)$ is not a winning strategy for Player 0. Thus we obtain a vertex $(\langle \mu p \psi \rangle, s_2)$ and finite play π_1 in $\mathcal{G}(\mathcal{A}(\psi), \mathcal{K}[p \mapsto S'], s_1)$ which is consistent with f such that $(\langle p \rangle, s_2)$ is the last vertex in π_1 , $s_2 \in S_\mu$ but $s_2 \notin S'$. Inductively we obtain an infinite sequence of vertices $(\langle p \rangle, s_i)_{i \in \omega}$ in game $\mathcal{G}(\mathcal{A}(\psi), \mathcal{K}[p \mapsto S'], s_1)$ consistent with restrictions of f . Hence, the following play in $\mathcal{G}(\mathcal{A}(\mu p \psi), \mathcal{K}, s_0)$ which is consistent with f and therefore won by Player 0:

$$\pi = (\langle \mu p \psi \rangle, s_0) \pi_0 (\langle \mu p \psi \rangle, s_1) \pi_1 (\langle \mu p \psi \rangle, s_2) \dots \quad (42)$$

Since $\Omega(\langle \mu p \psi \rangle)$ is the maximum priority of the automaton $\mathcal{A}(\mu p \psi)$ and it is odd thus we have a contradiction. Therefore, $(\langle \mu p \psi \rangle, s_0)$.

Case $\varphi = \nu p \psi$, similar to the previous case.

7 Model Checking & Satisfiability

Proposition 3 (*Model checking in μ -Calculus*) Given a finite pointed Kripke structure (\mathcal{K}, s_I) and an L_μ formula φ , determine whether $(\mathcal{K}, s_I) \models \varphi$.

Let (\mathcal{K}, s_I) be a pointed transition system and \mathcal{B} an alternating automaton. Let first define a parity game \mathcal{G} as a tuple as follows:

$$\mathcal{T} = (L_0, L_1, l_I, M, \Omega) \quad (43)$$

where

- $L_0 = V_0$ and $L_1 = V_1$ where V_0, V_1 are disjoint sets of vertices in arena $\mathcal{A} \in \text{mathcal{G}}$.
- $l_I = v_0$ where v_0 is the starting vertex in \mathcal{G} .
- $M \subseteq E$ where E are the edge relations among vertices of the game.
- $\Omega : (L_0 \cup L_1) \rightarrow \omega$ is a priority function over finite range ω .

Clearly, the ordered pair (L, M) where $L = L_0 \cup L_1$ is a directed graph, which is denoted $G(\mathcal{T})$ and called the game graph of \mathcal{T} .

Proposition 4 (*Model Checking Problem Reduction to Acceptance Problem*) The alternating tree automaton \mathcal{B} accepts (\mathcal{K}, s_I) if and only if Player 0 has a winning strategy in the parity game $\mathcal{G}(\mathcal{T}) = (\mathcal{K}, \mathcal{B}, s_I)$ (i.e., $\mathcal{T} = (V_0, V_1, q_I \times s_I, \Omega)$).

Proof sketch Just observe that accepting runs of \mathcal{B} on (\mathcal{S}, s_I) and a winning strategy trees for Player 0 in $\mathcal{G}(\mathcal{T}) = (\mathcal{K}, \mathcal{B}, s_I)$ are identical.

Proposition 5 (*Satisfiability in μ -Calculus*) Given an L_μ formula φ , determine whether there exists a pointed Kripke structure (\mathcal{K}, s_I) such that $\mathcal{K} \models \varphi$.

Solving the nonemptiness problem for alternating tree automata amounts to finding a tree that is accepted. Solving the winner problem for a parity game amounts to finding a memoryless winning strategy tree.

Proposition 6 (*Satisfiability Problem Encoding to Emptiness*) The automaton \mathcal{C} accepts a pointed Kripke structure (\mathcal{K}, s_I) if and only if Player 0 wins the game \mathcal{T} .

8 Complexity Bounds

Theorem 4 [7]

- The Model-Checking problem for μ -Calculus, is solvable in time:

$$\mathcal{O}(\ln(\frac{2nkn}{b})^{\lfloor b/2 \rfloor}) \quad (44)$$

where k is the number of worlds of the Kripke structure, l is the size of accessibility relation, n is the number of subformulas and b is the alternation depth.

- The model checking is in $UP \cap co-UP$.

Theorem 5 [7]

- The word problem for alternating tree automaton is solvable in time

$$\mathcal{O}(\ln(\frac{2nkn}{b})^{\lfloor b/2 \rfloor}) \quad (45)$$

where k is the number of worlds of the Kripke structure, l is the size of accessibility relation, n is the number of states of automaton and b is the index of the automaton.

- The acceptance is in $UP \cap co-UP$.

Proposition 7 [1] [7]

- The non-emptiness problem of alternating tree automaton is in $EXPTIME$.
- The satisfiability of μ -Calculus is in $EXPTIME$.

9 Conclusion

In this paper, we had introduced modal μ -Calculus and provides its equivalent translation into an alternating tree automaton which implies that every Kripke query recognizable by an alternating tree automaton can be defined by a modal μ -Calculus formula. We proposed how alternating tree automaton, together with parity game semantics provides a reasonable reduction from modal μ -Calculus model checking problem to the membership problem and from satisfiability problem to the emptiness problem with reasonable complexity bounds.

References

1. E. A. Emerson and C. S. Jutla. The complexity of tree automata and logics of programs. *Symposium on Foundations of Computer Science*, 0:328–337, 1988.
2. Erich Grädel, Wolfgang Thomas, and Thomas Wilke, editors. *Automata Logics, and Infinite Games: A Guide to Current Research*, Lecture Notes in Computer Science, New York, NY, USA, 2002. Springer-Verlag New York, Inc.
3. Daniel Kirsten. Alternating tree automata and parity games. In Grädel et al. [2], pages 153–167.
4. Panagiotis Manolios. *Mu-calculus model-checking*. Kluwer Academic Publishers, Norwell, MA, USA, 2000.
5. Rene Mazala. Infinite games. In Grädel et al. [2], pages 22–38.
6. Igor Walukiewicz. Monadic second-order logic on tree-like structures. *Theor. Comput. Sci.*, 275(1-2):311–346, 2002.
7. Thomas Wilke. Alternating tree automata, parity games, and modal μ -calculus. *Bulletin of the Belgian Math. Soc.*, 8(2):359–391, 2001.
8. Júlia Zappe. Modal μ -calculus and alternating tree automata. In Grädel et al. [2], pages 171–184.

10 Appendix

The notation $\mathcal{G} \downarrow v$ for a parity game $\mathcal{G} = ((V_0, V_1, E), \chi)$ and $v \in V_0 \cup V_1$ to denote the subgame game $\mathcal{G}[U]$, where U is the set of those vertices that are reachable from v .

Lemma 1 *Let (\mathcal{G}, v) be an initialized parity game, f a memoryless winning strategy for Player 0 in this game and π a play consistent with f . Let $v' = \pi(i)$ for some $i \in \omega$. Then the restriction of f to the vertices of $\mathcal{G} \downarrow v'$ is a winning strategy for Player 0 in the game $\mathcal{G} \downarrow v'$.*

Proof By assumption every play in \mathcal{G} starting in v' only visit vertices that are in $\mathcal{G} \downarrow v'$. In order to prove that play π' in $\mathcal{G} \downarrow v'$ consistent with f starting in v' , let proceed with prove by contradiction. Hence, there exists a play π'' which is consistent with f starting in v' but not won by Player 0. Now take the i th-prefix p , which is $p = \pi(0) \dots \pi(i-1)$. Since f is memoryless, the concatenated play $p\pi''$ is also consistent with f by adding finite prefix to infinite play. Thus, $p\pi''$ is not won by Player 0 since adding a prefix is invariant. Therefore a contradicting the assumption f being a winning strategy. Therefore, every play π is consistent with f starting in v is won by Player 0.

Lemma 2 *Let ψ_1, ψ_2 be L_μ formulas in normal form. Then the following is true:*

$$L(\mathcal{A}(\psi_1 \vee \psi_2)) = L(\mathcal{A}(\psi_1)) \cup L(\mathcal{A}(\psi_2)) \quad (46)$$

Proof \subseteq : Let $(\mathcal{K}, s_I) \in L(\mathcal{A}(\psi_1 \vee \psi_2))$. By the definition of alternating tree automata, there exists a memoryless winning strategy f for Player 0 in the parity game $\mathcal{G}(\psi_1 \vee \psi_2, \mathcal{K}, s_I)$ having initial vertex $(\langle \psi_1, \psi_2 \rangle, s_I)$. Hence, every consistent play starting in $(\langle \psi_1, \psi_2 \rangle, s_I)$ has form $(\langle \psi_1, \psi_2 \rangle, s_I)(\langle \psi_i \rangle, s_I)\pi$ for some $i \in \{0, 1\}$ and some play π . By Lemma-1, there is a winning strategy for Player 0 in game $\mathcal{G}(\psi_1 \vee \psi_2, \mathcal{K}, s_I) \downarrow (\langle \psi_i \rangle, s_I) = \mathcal{G}(\psi_i, \mathcal{K}, s_I)$. Therefore, we have $\mathcal{K}, f_I \in L(\mathcal{A}(\psi_i)) \subseteq L(\mathcal{A}(\psi_1)) \cup L(\mathcal{A}(\psi_2))$.

\supseteq : Let $(\mathcal{K}, s_I) \in L(\mathcal{A}(\psi_1)) \cup L(\mathcal{A}(\psi_2))$. The Player 0 has a memoryless winning strategy f_i in a parity game $\mathcal{G}(\psi_i, \mathcal{K}, s_I)$ for some $i \in \{1, 2\}$. By extending the strategy f_i to a strategy f of Player 0 in $\mathcal{G}(\psi_1 \vee \psi_2, \mathcal{K}, s_I)$ by following:

$$f(v) = \begin{cases} \{f_i(v) & \text{if } v \text{ is } 0\text{-vertex in } \mathcal{G}(\psi_i, \mathcal{K}, s_I) \\ (\langle \psi_i \rangle, s_I) & \text{if } v = (\langle \psi_1 \vee \psi_2 \rangle, s_I) \end{cases} \quad (47)$$

$f(v)$ is a strategy for Player 0. Hence, by definition of f , any play $(\langle \psi_1, \psi_2 \rangle, s_I)(\langle \psi_i \rangle, s_I)\pi$ for some $i \in \{0, 1\}$ and some play π is consistent with f . Thus suffix $\langle \psi_i \rangle\pi$ is consistent with f_i and therefore won by Player 0. By lemma-1, $(\langle \psi_1, \psi_2 \rangle, s_I)(\langle \psi_i \rangle, s_I)\pi$ is won by Player 0. Therefore, f is a winning strategy for Player 0 in $\mathcal{G}(\psi_1 \vee \psi_2, \mathcal{K}, s_I)$ implies $(\mathcal{K}, s_I) \in L(\mathcal{A}(\psi_1 \vee \psi_2))$.

Lemma 3 *Let ψ_1, ψ_2 be L_μ formulas in normal form. Then the following is true:*

$$L(\mathcal{A}(\psi_1 \wedge \psi_2)) = L(\mathcal{A}(\psi_1)) \cap L(\mathcal{A}(\psi_2)) \quad (48)$$

Proof Given in [8].

Lemma 4 *Let ψ be an L_μ formula in normal form and transition system $\mathcal{K} = \{S, E, \lambda\}$. Then the following is true:*

$$L(\mathcal{A}(\Box\psi)) = \{(\mathcal{K}, s_I) \mid \forall s' \in sR : (\mathcal{K}, s') \in L(\mathcal{A}(\psi))\} \quad (49)$$

Proof \subseteq : let $(\mathcal{K}, s_I) \in L(\mathcal{A}(\Box\psi))$ and $\mathcal{K} = \{S, E, \lambda\}$. There is a memoryless winning strategy f for Player 0 in game $\mathcal{G}(\mathcal{A}(\Box\psi), \mathcal{K}, s_I)$. The initial vertex $(\langle\Box\psi\rangle, s_I)$ has for each $s' \in sR$ a successor $(\langle\psi\rangle, s')$. As f is a winning strategy for Player 0, thus for prefix $(\langle\Box\psi\rangle, s_I)(\langle\psi\rangle, s')$ is consistent with f for all $s' \in sR$. By Lemma-1, there is winning strategy for Player 0 in $\mathcal{G}(\mathcal{A}(\psi), \mathcal{K}, s_I) = \mathcal{G}(\mathcal{A}(\Box\psi), \mathcal{K}, s_I) \downarrow (\langle\psi\rangle, s')$ for all $s' \in sR$. Therefore, for all $s' \in sR$ we have $(\mathcal{K}, s') \in L(\mathcal{A}(\psi))$.

\supseteq : For every $s' \in sR$, there is a winning strategy in $f^{s'} : V^{s'} \times V_0^{s'} \rightarrow V^{s'}$ where $V^{s'}$ and $V_0^{s'}$ denote the set of vertices in game $\mathcal{G}(\mathcal{A}(\psi), \mathcal{K}, s_I)$. Now consider the game $\mathcal{G}(\mathcal{A}(\Box\psi), \mathcal{K}, s_I)$ with V and V_0 being its vertices where its initial vertex is $(\langle\Box\psi\rangle, s_I)$ and its successor are $\{(\psi, s') | \forall s' \in s_I R\}$. By combining strategies $f^{s'}$ and f for play π we have:

$$f(\pi) = \left\{ \begin{array}{l} f^{s'}((\langle\psi\rangle, s'), \pi') \text{ if } \pi = ((\langle\psi\rangle, s))(\langle\psi\rangle, s')\psi' \text{ for some } s' \in sR, \pi' \in (V^{s'})^*. \end{array} \right. \quad (50)$$

Thus, $f^{s'}$ is a winning strategy for Player 0 in the game $\mathcal{G}(\mathcal{A}(\psi), \mathcal{K}, s') = \mathcal{G}(\mathcal{A}(\Box\psi), \mathcal{K}, s) \downarrow (\langle\psi\rangle, s')$ implies that f is a winning strategy for Player 0 in the game $\mathcal{G}(\mathcal{A}(\Box\psi), \mathcal{K}, s)$. Therefore, $(\mathcal{K}, s_I) \in L(\mathcal{A}(\Box\psi))$.

Lemma 5 *Let ψ be an L_μ formula in normal form and transition system $\mathcal{K} = \{S, E, \lambda\}$. Then the following is true:*

$$L(\mathcal{A}(\Diamond\psi)) = \{(\mathcal{K}, s_I) | \exists s' \in sR : (\mathcal{K}, s') \in L(\mathcal{A}(\psi))\} \quad (51)$$

Proof Given in [8].